

Relación de Ejercicios de Clases y Objetos (4): Fracciones

Crear la clase Fracción que nos permite representar fracciones y hacer operaciones con ellas. La clase contará con los siguientes elementos:

- Atributos:
 - **numerador** y **denominador**, que serán *int*.
- Constructores:
 - Un constructor al que le pasamos dos enteros (que corresponderán al numerador y denominador) y que nos crea la fracción tal cual. Si el denominador es 0 lanzará una excepción.
 - Otro constructor al que le pasamos un *int* y nos lo representará como fracción (lo ponemos dividido entre 1).
 - Otro constructor al que le pasamos un *double* y que nos lo representará como fracción (el método es muy simple: ponemos el número en el numerador, un 1 en el denominador, y vamos multiplicando ambos por 10 hasta que no haya decimales).
- Propiedades:
 - **getNumerador** y **setNumerador**, para consultar o modificar el numerador.
 - **getDenominador** y **setDenominador**, para consultar o modificar el denominador. Si intentamos guardar el valor 0 en el denominador, dará una excepción.
- Métodos privados:
 - Os van a hacer falta el mcm y el MCD, así que podéis hacer un par de métodos privados para poder usarlos desde el resto de métodos.
- Métodos:
 - **simplificar()**, que nos simplifica una fracción. Se hace dividiendo numerador y denominador entre el máximo común divisor de ambos.
 - **toString()**, que nos devolverá un *String* con la fracción (p.ej.: si el numerador es 3 y el denominador 4, nos devolverá: "3/4").
- Métodos (Operadores):
 - Fraccion **negate()**: nos devolverá nuestra fracción en negativo (p.ej.: si nuestra fracción es 3/4, devolverá la fracción -3/4 y viceversa).
 - Fraccion **add**(Fraccion f): sumará nuestra fracción con la que le pasamos por parámetro y nos devolverá una fracción con el resultado de la suma.
 - Fraccion **subtract**(Fraccion f): hará la resta de nuestra fracción menos la fracción que le pasamos por parámetro y devolverá el resultado.
 - Fraccion **multiply**(Fraccion f): multiplicará nuestra fracción por la que le pasamos por parámetro y devolverá el resultado.
 - Fraccion **divide**(Fraccion f): hará la división de nuestra fracción entre la que le pasamos por parámetro y devolverá el resultado.

- boolean **equals**(Fraccion f): nos devolverá *true* si el valor de nuestra fracción y la que le pasamos por parámetro es el mismo (si le pasamos dos fracciones distintas pero con el mismo valor, p.ej.: $1/2$ y $2/4$, debería dar *true*).
- int **compareTo**(Fraccion f): nos comparará nuestra fracción con la fracción que le pasamos por parámetro y nos devolverá un entero con el resultado de la comparación:
 - Si nuestra fracción es menor que la otra, devolveremos -1.
 - Si nuestra fracción es mayor que la otra, devolveremos 1.
 - Si son iguales, devolveremos 0.

En el **main**, hacer un menú que nos permita probar todas las funciones.